Multiclass Data Segmentation using Diffuse Interface Methods on Graphs

Cristina Garcia-Cardona, Ekaterina Merkurjev, Andrea L. Bertozzi, Arjuna Flenner and Allon G. Percus

***Abstract*—We present two graph-based algorithms for multiclass segmentation of high-dimensional data on graphs. The algorithms use a diffuse interface model based on the Ginzburg-Landau functional, related to total variation and graph cuts. A multiclass extension is introduced using the Gibbs simplex, with the functional's double-well potential modified to handle the multiclass case. The first algorithm minimizes the functional using a convex splitting numerical scheme. The second algorithm uses a graph adaptation of the classical numerical Merriman-Bence-Osher (MBO) scheme, which alternates between diffusion and thresholding. We demonstrate the performance of both algorithms experimentally on synthetic data, image labeling, and several benchmark data sets such as MNIST, COIL and WebKB. We also make use of fast numerical solvers for finding the eigenvectors and eigenvalues of the graph Laplacian, and take advantage of the sparsity of the matrix. Experiments indicate that the results are competitive with or better than the current state-of-the-art in multiclass graph-based segmentation algorithms for high-dimensional data.**

***Index Terms*—segmentation, Ginzburg-Landau functional, diffuse interface, MBO scheme, graphs, convex splitting, image processing, high-dimensional data.**

I. Introduction

Multiclass segmentation is a fundamental problem in machine learning. In this paper, we present a general approach to multiclass segmentation of high-dimensional data on graphs, motivated by the diffuse interface model in [4]. The method applies L_2 gradient flow minimization of the Ginzburg-Landau (GL) energy to the case of functions defined on graphs.

The GL energy is a smooth functional that converges, in the limit of a vanishing interface width, to the total variation (TV) [5], [44]. There is a close connection between TV minimization and graph cut minimization. Given a graph $G = (V, E)$ with vertex set V, edge set E, and edge weights w_{ij} for $i, j \in V$, the TV norm of a function f on V is

$$||f||_{TV} = \frac{1}{2} \sum_{i,j \in V} w_{ij} |f_i - f_j|. \tag{1}$$

If f_i is interpreted as a classification of vertex i, minimizing TV is exactly equivalent to minimizing the graph cut. TV-based methods have recently been used [8], [9], [57] to find good approximations for normalized graph cut minimization, an NP-hard problem. Unlike methods such as spectral clustering, normalized TV minimization provides a tight relaxation of the problem, though cannot usually be solved exactly. The approach in [4] performs binary segmentation on graphs by using the GL functional as a smooth but arbitarily close approximation to the TV norm.

Our new formulation builds on [4], using a semi-supervised learning (SSL) framework for multiclass graph segmentation. We employ a phase-field representation of the GL energy functional: a vector-valued quantity is assigned to every node on the graph, such that each of its components represents the fraction of the phase, or class, present in that particular node. The components of the field variable add up to one, so the phase-field vector is constrained to lie on the Gibbs simplex. The phase-field representation, used in material science to study the evolution of multi-phase systems [32], has been studied previously for multiclass image segmentation [47]. Likewise, the simplex idea has been used for image segmentation [13], [37]. However, to the best of our knowledge, our diffuse interface approach is the first application of a vector-field GL representation to the general problem of multiclass semi-supervised classification of high-dimensional data on graphs.

In addition, we apply this Gibbs simplex idea to the graph-based Merriman-Bence-Osher (MBO) scheme developed in [49]. The MBO scheme [50] is a well-established PDE method for evolving an interface by mean curvature. As with the diffuse interface model, tools for nonlocal calculus [33] are used in [49] to generalize the PDE formulation to the graph setting. By introducing the phase-field representation to the graph-based MBO scheme, we develop another new and highly efficient algorithm for multiclass segmentation in a SSL framework.

The main contributions of our work are therefore twofold. First, we introduce two new graph-based methods for multiclass data segmentation, namely a multiclass GL minimization method based on the binary representation described in [4] and a multiclass graph-based MBO method motivated by the model in [49]. Second, we present very efficient algorithms derived from these methods, and applicable to general multiclass high-dimensional data segmentation.

The paper is organized as follows. In section II, we discuss prior related work, as well as motivation for the methods proposed here. We then describe our two new multiclass algorithms in section III (one in section III-A and one in III-B). In section IV, we present experimental results on benchmark data sets, demonstrating the effectiveness of our methods. Finally, in section V, we conclude and discuss ideas for future work.

C. Garcia-Cardona and A.G. Percus are with the Institute of Mathematical Sciences at Claremont Graduate University. E-mail: cristina.cgarcia@gmail.com, allon.percus@cgu.edu.

E. Merkurjev and A.L. Bertozzi are with the Department of Mathematics at University of California, Los Angeles. Email: {kmerkurev,bertozzi}@math.ucla.edu.

A. Flenner is with the Naval Air Warfare Center, China Lake.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
2014		**00-00-2014 to 00-00-2014**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Multiclass Data Segmentation using Diffuse Interface Methods on Graphs	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **University of California Los Angeles (UCLA),Department of Mathematics,520 Portola Plaza,Los Angeles,CA,90095**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
accepted in IEEE PAMI, 2014.

14. ABSTRACT

We present two graph-based algorithms for multiclass segmentation of high-dimensional data on graphs. The algorithms use a diffuse interface model based on the Ginzburg-Landau functional, related to total variation and graph cuts. A multiclass extension is introduced using the Gibbs simplex, with the functional?s double-well potential modified to handle the multiclass case. The first algorithm minimizes the functional using a convex splitting numerical scheme. The second algorithm uses a graph adaptation of the classical numerical Merriman-Bence-Osher (MBO) scheme, which alternates between diffusion and thresholding. We demonstrate the performance of both algorithms experimentally on synthetic data, image labeling, and several benchmark data sets such as MNIST, COIL and WebKB. We also make use of fast numerical solvers for finding the eigenvectors and eigenvalues of the graph Laplacian, and take advantage of the sparsity of the matrix. Experiments indicate that the results are competitive with or better than the current state-of-the-art in multiclass graph-based segmentation algorithms for high-dimensional data.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			
unclassified	**unclassified**	**unclassified**	**Same as Report (SAR)**	**14**	

II. Previous Work

A. General Background

In this section, we present prior related work, as well as specific algorithms that serve as motivation for our new multiclass methods.

The discrete graph formulation of GL energy minimization is an example of a more general form of energy (or cost) functional for data classification in machine learning,

$$E(\psi) = R(\psi) + \mu||\psi - \hat{\psi}||, \tag{2}$$

where ψ is the classification function, $R(\psi)$ is a regularization term, and $||\psi - \hat{\psi}||$ is a fidelity term, incorporating most (supervised) or just a few (semi-supervised) of the known values $\hat{\psi}$. The choice of R has non-trivial consequences in the final classification accuracy. In instances where $||\cdot||$ is the L_2 norm, the resulting cost functional is a tradeoff between accuracy in the classification of given labels and function smoothness. It is desirable to choose R to preserve the sharp discontinuities that may arise in the boundaries between classes. Hence the interest in formulations that can produce piecewise constant solutions [7].

Graph-based regularization terms, expressed by way of the discrete Laplace operator on graphs, are often used in semi-supervised learning as a way to exploit underlying similarities in the data set [3], [14], [61], [66]–[68]. Additionally, some of these methods use a matrix representation to apply eq. (2) to the multiple-class case [14], [61], [66], [68]. The rows in the matrix correspond to graph vertices and the columns to indicator functions for class membership: the class membership for vertex i is computed as the column with largest component in the ith row. The resulting minimization procedure is akin to multiple relaxed binary classifications running in parallel. This representation is different from the Gibbs simplex we use, as there is usually no requirement that the elements in the row add up to 1. An alternative regularization method for the graph-based multiclass setup is presented in [56], where the authors minimize a Kullback-Leibler divergence function between discrete probability measures that translates into class membership probabilities.

Not all the methods deal directly with the multiple classes in the data set. A different approach is to reduce the multiclass case to a series of two-class problems and to combine the sequence of resulting sub-classifications. Strategies employed include recursive partitioning, hierarchical classification and binary encodings, among others. For example, Dietterich and Bakiri use a binary approach to encode the class labels [22]. In [39], a pairwise coupling is described, in which each two-class problem is solved and then a class decision is made combining the decisions of all the subproblems. Szlam and Bresson present a method involving Cheeger cuts and split Bregman iteration [34] to build a recursive partitioning scheme in which the data set is repeatedly divided until the desired number of classes is reached. The latter scheme has been extended to mutliclass versions. In [10], a multiclass algorithm for the transductive learning problem in high-dimensional data classification, based on ℓ^1 relaxation of the Cheeger cut and the piecewise constant Mumford-Shah or Potts models, is described. Recently, a new TV-based method for multiclass clustering has been introduced in [9].

Our methods, on the other hand, have roots in the continuous setting as they are derived via a variational formulation. Our first method comes from a variational formulation of the L_2 gradient flow minimization of the GL functional [4], but which in a limit turns into TV minimization. Our second method is built upon the MBO classical scheme to evolve interfaces by mean curvature [50]. The latter has connections with the work presented in [26], where an MBO-like scheme is used for image segmentation. The method is motivated by the propagation of the Allen-Cahn equation with a forcing term, obtained by applying gradient descent to minimize the GL functional with a fidelity term.

Alternative variational principles have also been used for image segmentation. In [47], a multiclass labeling for image analysis is carried out by a multidimensional total variation formulation involving a simplex-constrained convex optimization. In that work, a discretization of the resulting PDEs is used to solve numerically the minimization of the energy. Also, in [13] a partition of a continuous open domain in subsets with minimal perimeter is analyzed. A convex relaxation procedure is proposed and applied to image segmentation. In these cases, the discretization corresponds to a uniform grid embedded in the Euclidean space where the domain resides. Similarly, diffuse interface methods have been used successfully in image impainting [6], [23] and image segmentation [26].

While our algorithms are inspired by continuous processes, they can be written directly in a discrete combinatorial setting defined by the graph Laplacian. This has the advantage, noted by Grady [37], of avoiding errors that could arise from a discretization process. We represent the data as nodes in a weighted graph, with each edge assigned a measure of similarity between the vertices it is connecting. The edges between nodes in the graph are not the result of a regular grid embedded in an Euclidean space. Therefore, a nonlocal calculus formulation [33] is the tool used to generalize the continuous formulation to a (nonlocal) discrete setting given by functions on graphs. Other nonlocal formulations for weighted graphs are included in [25], while [35] constitutes a comprehensive reference about techniques to cast continuous PDEs in graph form.The approach of defining functions with domains corresponding to nodes in a graph has successfully been used in areas, such as spectral graph theory [16], [51].

Graph-based formulations have been used extensively for image processing applications [7], [18], [19], [25], [36]–[38], [48], [54]. Interesting connections between these different algorithms, as well as between continuous and discrete optimizations, have been established in the literature. Grady has proposed a random walk algorithm [37] that performs interactive image segmentation using the solution to a combinatorial Dirichlet problem. Elmoataz et al. have developed generalizations of the graph Laplacian [25] for image denoising and manifold smoothing. Couprie et al. in [18] define a conveniently parameterized graph-based energy function that is able to unify graph cuts, random walker, shortest paths and watershed optimizations. There, the authors test different seeded image segmentation algorithms, and discuss possibilities to

optimize more general models with applications beyond image segmentation. In [19], alternative graph-based formulations of the continuous max-flow problem are compared, and it is shown that not all the properties satisfied in the continuous setting carry over to the discrete graph representation. For general data segmentation, Bresson et al. in [8], present rigorous convergence results for two algorithms that solve the relaxed Cheeger cut minimization, and show a formula that gives the correspondence between the global minimizers of the relaxed problem and the global minimizers of the combinatorial problem. In our case, the convergence property of GL to TV has been known to hold in the continuum [44], but has recently been shown in the graph setting as well [5].

B. Binary Segmentation using the Ginzburg-Landau Functional

The classical Ginzburg-Landau (GL) functional can be written as:

$$GL(u) = \frac{\epsilon}{2}\int |\nabla u|^2 dx + \frac{1}{\epsilon}\int \Phi(u)dx, \tag{3}$$

where u is a scalar field defined over a space of arbitrary dimensionality and representing the state of the phases in the system, ∇ denotes the spatial gradient operator, $\Phi(u)$ is a double-well potential, such as $\Phi(u) = \frac{1}{4}(u^2-1)^2$, and ϵ is a positive constant. The two terms are: a smoothing term that measures the differences in the components of the field, and a potential term that measures how far each component is from a specific value (± 1 in the example above). In the next subsection, we derive the proper formulation in a graph setting.

It is shown in [44] that the $\epsilon \to 0$ limit of the GL functional, in the sense of Γ-convergence, is the Total Variation (TV) semi-norm:

$$GL(u) \to_\Gamma ||u||_{TV}. \tag{4}$$

Due to this relationship, the two functionals can sometimes be interchanged. The advantage of the GL functional is that its L_2 gradient flow leads to a linear differential operator, which allows us to use fast methods for minimization.

Equation (3) arises in its continuum form in several imaging applications including inpainting [6] and segmentation [26]. In such problems, one typically considers a gradient flow in which the continuum Laplacian is most often discretized in space using the 4-regular graph. The inpainting application in [6] considers a gradient flow in an H^{-1} inner product resulting in the biharmonic operator which can be discretized by considering two applications of the discrete Laplace operator. The model in (3) has also been generalized to wavelets [23], [24] by replacing the continuum Laplacian with an operator that has eigenfunctions specified by the wavelet basis. Here we consider a general graphical framework in which the graph Laplacian replaces the continuum Laplace operator.

We also note that the standard practice in all of the examples above is to introduce an additional term in the energy functional to escape from trivial steady-state solutions (e.g., all labels taking on the same value). This leads to the expression

$$E(u) = GL(u) + F(u, \hat{u}), \tag{5}$$

where F is the additional term, usually called *fidelity*. This term allows the specification of any known information, for example, regions of an image that belong to a certain class.

Inspired in part by the PDE-based imaging community, where variational algorithms combining ideas from spectral methods on graphs with nonlinear edge detection methods are common [33], Bertozzi and Flenner extended in [4] the L_2 gradient flow of the Ginzburg-Landau (GL) energy functional to the domain of functions on a graph.

The energy $E(u)$ in (5) can be minimized in the L_2 sense using gradient descent. This leads to the following dynamic equation (*modified Allen-Cahn equation*):

$$\frac{\partial u}{\partial t} = -\frac{\delta GL}{\delta u} - \mu\frac{\delta F}{\delta u} = \epsilon\Delta u - \frac{1}{\epsilon}\Phi'(u) - \mu\frac{\delta F}{\delta u} \tag{6}$$

where Δ is the Laplacian operator. A local minimizer is obtained by evolving this expression to steady state. Note that E is not convex, and may have multiple local minima.

Before continuing further, let us introduce some graph concepts that we will use in subsequent sections.

1) Graph Framework for Large Data Sets: Let G be an undirected graph $G = (V, E)$, where V and E are the sets of vertices and edges, respectively. The vertices are the building blocks of the data set, such as points in $\mathbb{R}^n$ or pixels in an image. The similarity between vertices i and j is measured by a weight function $w(i, j)$ that satisfies the symmetric property $w(i, j)$= $w(j, i)$. A large value of $w(i, j)$ indicates that vertices i and j are similar to each other, while a small $w(i, j)$ indicates that they are dissimilar. For example, an often used similarity measure is the Gaussian function

$$w(i,j) = \exp\left(-\frac{d(i,j)^2}{\sigma^2}\right), \tag{7}$$

with $d(i, j)$ representing the distance between the points associated with vertices i and j, and σ^2 a positive parameter.

Define $\mathbf{W}$ as the matrix $W_{ij} = w(i, j)$, and define the degree of a vertex $i \in V$ as

$$d_i = \sum_{j\in V} w(i,j). \tag{8}$$

If $\mathbf{D}$ is the diagonal matrix with elements d_i, then the graph Laplacian is defined as the matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$.

2) Ginzburg-Landau Functional on Graphs: The continuous GL formulation is generalized to the case of weighted graphs via the graph Laplacian. Nonlocal calculus, such as that outlined in [33], shows that the Laplace operator is related to the graph Laplacian matrix defined above, and that the eigenvectors of the discrete Laplacian converge to the eigenvectors of the Laplacian [4]. However, to guarantee convergence to the continuum differential operator in the limit of large sample size, the matrix $\mathbf{L}$ must be correctly scaled [4]. Although several versions exist, we use the symmetric normalized Laplacian

$$\mathbf{L_s} = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}. \tag{9}$$

since its symmetric property allows for more efficient implementations. Note that $\mathbf{L_s}$ satisfies:

$$\langle \boldsymbol{u}, \mathbf{L_s}\boldsymbol{u}\rangle = \frac{1}{2}\sum_{i,j} w(i,j)\left(\frac{u_i}{\sqrt{d_i}} - \frac{u_j}{\sqrt{d_j}}\right)^2 \tag{10}$$

for all $\boldsymbol{u} \in \mathbb{R}^n$. Here the subscript i refers to the i^{th} coordinate of the vector, and the brackets denote the standard dot product. Note also that $\mathbf{L_s}$ has nonnegative, real-valued eigenvalues.

Likewise, it is important to point out that for tasks such as data classification, the use of a graphs has the advantage of providing a way to deal with nonlinearly separable classes as well as simplifying the processing of high dimensional data.

The GL functional on graphs is then expressed as

$$GL(\boldsymbol{u}) = \frac{\epsilon}{2}\langle \boldsymbol{u}, \mathbf{L_s}\boldsymbol{u}\rangle + \frac{1}{4\epsilon}\sum_{i\in V}\left(u_i^2 - 1\right)^2, \qquad (11)$$

where u_i is the (real-valued) state of node i. The first term replaces the gradient term in (3), and the second term is the double-well potential, appropriate for binary classifications.

3) Role of Diffuse Interface Parameter ϵ: In the minimization of the GL functional, two conflicting requirements are balanced. The first term tries to maintain a smooth state throughout the system, while the second term tries to force each node to adopt the values corresponding to the minima of the double-well potential function. The two terms are balanced through the diffuse interface parameter ϵ.

Recall that in the continuous case, it is known that the GL functional (*smoothing + potential*) converges to total variation (TV) in the limit where the diffuse interface parameter $\epsilon \to 0$ [44]. An analogous property has recently been shown in the case of graphs as well, for binary segmentations [5]. Since TV is an L_1-based metric, TV-minimization leads to sparse solutions, namely indicator functions that closely resemble the discrete solution of the original NP-hard combinatorial segmentation problem [9], [57]. Thus, the GL functional actually becomes an L_1 metric in the small ϵ limit, and leads to sharp transitions between classes. Intuitively, the convergence of GL to TV holds because in the limit of a vanishing interface, the potential takes precedence and the graph nodes are forced towards the minima of the potential, achieving a configuration of minimal length of transition. This is contrast to more traditional spectral clustering approaches, which can be understood as L_2-based methods and do not favor sparse solutions. Furthermore, while the smoothness of the transition in the GL functional is regulated by ϵ, in practice the value of ϵ does not have to be decreased all the way to zero to obtain sharp transitions (an example of this is shown later in Figure 4). This capability of modeling the separation of a domain into regions or phases with a controlled smoothness transition between them makes the diffuse interface description attractive for segmentation problems, and distinguishes it from more traditional graph-based spectral partitioning methods.

4) Semi-Supervised Learning (SSL) on Graphs: In graph-based learning methods, the graph is constructed such that the edges represent the similarities in the data set and the nodes have an associated real state that encodes, with an appropriate thresholding operation, class membership.

In addition, in some data sets, the label of a small fraction of data points is known beforehand. This considerably improves the learning accuracy, explaining in part the popularity of semi-supervised learning methods. The graph generalization of the diffuse interface model handles this condition by using the labels of known points. The GL functional for SSL is:

$$\begin{aligned} E(\boldsymbol{u}) &= \frac{\epsilon}{2}\langle \boldsymbol{u}, \mathbf{L_s}\boldsymbol{u}\rangle + \frac{1}{4\epsilon}\sum_{i\in V}\left(u_i^2 - 1\right)^2 \\ &\quad + \sum_{i\in V}\frac{\mu_i}{2}\left(u_i - \hat{u}_i\right)^2. \end{aligned} \qquad (12)$$

The final term in the sum is the new fidelity term that enforces label values that are known beforehand. μ_i is a parameter that takes the value of a positive constant μ if i is a fidelity node and zero otherwise, and $\hat{u}_i$ is the known value of fidelity node i. This constitutes a soft assignment of fidelity points: these are not fixed but allowed to change state.

Note that since GL does not guarantee searching in a space orthogonal to the trivial minimum, alternative constraints could be introduced to obtain partitioning results that do not depend on fidelity information (unsupervised). For example, a mass-balance constraint, $\boldsymbol{u} \perp \mathbf{1}$, has been used in [4] to insure solutions orthogonal to the trivial minimum.

C. MBO Scheme for Binary Classification

In [50], Merriman, Bence and Osher propose alternating between the following two steps to approximate motion by mean curvature, or motion in which normal velocity equals mean curvature:

1) *Diffusion.* Let $u^{n+\frac{1}{2}} = S(\delta t)u^n$ where $S(\delta t)$ is the propagator (by time δt) of the standard heat equation:
$$\frac{\partial u}{\partial t} = \Delta u. \qquad (13)$$
2) *Thresholding.* Let
$$u^{n+1} = \begin{cases} 1 & \text{if } u^{n+\frac{1}{2}} \geq 0, \\ -1 & \text{if } u^{n+\frac{1}{2}} < 0. \end{cases}$$

This MBO scheme has been rigorously proven to approximate motion by mean curvature by Barles [2] and Evans [27] .

The algorithm is related to solving the basic (unmodified) Allen-Cahn equation, namely equation (6) without the fidelity term. If we consider a time-splitting scheme (details in [26]) to evolve the equation, in the $\epsilon \to 0$ limit, the second step is simply thresholding [50]. Thus, as $\epsilon \to 0$, the time splitting scheme above consists of alternating between diffusion and thresholding steps (MBO scheme mentioned above). In fact, it has been shown [53] that in the limit $\epsilon \to 0$, the rescaled solutions $u_\epsilon(z, t/\epsilon)$ of the Allen-Cahn equation yield motion by mean curvature of the interface between the two phases of the solutions, which the MBO scheme approximates.

The motion by mean curvature of the scheme can be generalized to the case of functions on a graph in much the same way as the procedure followed for the modified Allen-Cahn equation (6) in [4]. Merkurjev et al. have pursued this idea in [49], where a modified MBO scheme on graphs has been applied to the case of binary segmentation. The motivation comes from [26] by Esedoglu and Tsai, who propose threshold dynamics for the two-phase piecewise constant Mumford-Shah (MS) functional. The authors derive the scheme by applying a two-step time splitting scheme to the gradient descent equation resulting from the minimization of the MS functional, so that

the second step is the same as the one in the original MBO scheme. Merkurjev et al. in [49] also apply a similar time splitting scheme, but now to (6). The Δu term is then replaced with a more general graph term $-\mathbf{L_s}\boldsymbol{u}$. The discretized version of the algorithm is:

1) Heat equation with forcing term:

$$\frac{\boldsymbol{u}^{n+\frac{1}{2}} - \boldsymbol{u}^n}{dt} = -\mathbf{L_s}\boldsymbol{u}^n - \boldsymbol{\mu}(\boldsymbol{u}^n - \hat{\boldsymbol{u}}). \qquad (14)$$

2) Thresholding:

$$u_i^{n+1} = \begin{cases} 1, & \text{if } u_i^{n+\frac{1}{2}} > 0, \\ -1, & \text{if } u_i^{n+\frac{1}{2}} < 0. \end{cases}$$

Here, after the second step, u_i^n can take only two values of 1 or -1; thus, this method is appropriate for binary segmentation. The fidelity term scaling can be different from the one in (6).

The following section describes the modifications introduced to generalize this functional to multiclass segmentation.

III. Multiclass Data Segmentation

The main point of this paper is to show how to extend prior work to the multiclass case. This allows us to tackle a broad class of machine learning problems.

We use the following notation in the multiclass case. Given N_D data points, we generalize the label vector $\boldsymbol{u}$ to a label matrix $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_{N_D})^T$. Rather than node i adopting a single state $u_i \in \mathbb{R}$, it now adopts a composition of states expressed by a vector $\mathbf{u}_i \in \mathbb{R}^K$ where the kth component of $\mathbf{u}_i$ is the strength with which it takes on class k. The matrix $\mathbf{U}$ has dimensions $N_D \times K$, where K is the total number of possible classes.

For each node i, we require the vector $\mathbf{u}_i$ to be an element of the Gibbs simplex Σ^K, defined as

$$\Sigma^K := \left\{ (x_1, \ldots, x_K) \in [0,1]^K \;\middle|\; \sum_{k=1}^{K} x_k = 1 \right\}. \qquad (15)$$

Vertex k of the simplex is given by the unit vector $\boldsymbol{e}_k$, whose kth component equals 1 and all other components vanish. These vertices correspond to pure phases, where the node belongs exclusively to class k. The simplex formulation has a probabilistic interpretation, with $\mathbf{u}_i$ representing the probability distribution over the K classes. In other segmentation algorithms, such as spectral clustering, these real-valued variables can have different interpretations that are exploited for specific applications, as discussed in [38], [48].

A. Multiclass Ginzburg-Landau Approach

The multiclass GL energy functional for the phase field approach on graphs is written as:

$$\begin{aligned} E(\mathbf{U}) &= \frac{\epsilon}{2}\langle \mathbf{U}, \mathbf{L_s}\mathbf{U}\rangle + \frac{1}{2\epsilon}\sum_{i\in V}\left(\prod_{k=1}^{K}\frac{1}{4}\left\|\mathbf{u}_i - \boldsymbol{e}_k\right\|_{L_1}^2\right) \\ &\quad + \sum_{i\in V}\frac{\mu_i}{2}\left\|\mathbf{u}_i - \hat{\mathbf{u}}_i\right\|^2, \end{aligned} \qquad (16)$$

where

$$\langle \mathbf{U}, \mathbf{L_s}\mathbf{U}\rangle = \operatorname{trace}(\mathbf{U}^T\mathbf{L_s}\mathbf{U}),$$

and $\hat{\mathbf{u}}_i$ is a vector indicating prior class knowledge of sample i. We set $\hat{\boldsymbol{u}}_i = \boldsymbol{e}_k$ if node i is known to be in class k.

The first (smoothing) term in the GL functional (16) measures variations in the vector field. The simplex representation has the advantage that, like in Potts-based models but unlike in some other multiclass methods, the penalty assigned to differently labeled neighbors is independent of the integer ordering of the labels. The second (potential) term drives the system closer to the vertices of the simplex. For this term, we adopt an L_1 norm to prevent the emergence of an undesirable minimum at the center of the simplex, as would occur with an L_2 norm for large K. The third (fidelity) term enables the encoding of *a priori* information.

Note that one can obtain meaningful results without fidelity information (unsupervised), but the methods for doing so are not as straightforward. One example is a new TV-based modularity optimization method [41] that makes no assumption as to the number of classes and can be recast as GL minimization. Also, while Γ-convergence to TV in the graph setting has been proven for the binary segmentation problem [5], no similar convergence property has yet been proven for the multiclass case. We leave this as an open conjecture.

Following [4], we use a convex splitting scheme to minimize the GL functional in the phase field approach. The energy functional (16) is decomposed into convex and concave parts:

$$\begin{aligned} E(\mathbf{U}) &= E_{\text{convex}}(\mathbf{U}) + E_{\text{concave}}(\mathbf{U}) \\ E_{\text{convex}}(\mathbf{U}) &= \frac{\epsilon}{2}\langle \mathbf{U}, \mathbf{L_s}\mathbf{U}\rangle + \frac{C}{2}\langle \mathbf{U}, \mathbf{U}\rangle \\ E_{\text{concave}}(\mathbf{U}) &= \frac{1}{2\epsilon}\sum_{i\in V}\prod_{k=1}^{K}\frac{1}{4}\left\|\mathbf{u}_i - \boldsymbol{e}_k\right\|_{L_1}^2 \\ &\quad + \sum_{i\in V}\frac{\mu_i}{2}\left\|\mathbf{u}_i - \hat{\mathbf{u}}_i\right\|_{L_2}^2 - \frac{C}{2}\langle \mathbf{U}, \mathbf{U}\rangle \end{aligned}$$

with $C \in \mathbb{R}$ denoting a constant that is chosen to guarantee the convexity/concavity of the energy terms. Evaluating the second derivative of the partitions, and simplifying terms, yields:

$$C \geq \mu + \frac{1}{\epsilon}. \qquad (17)$$

The convex splitting scheme results in an unconditionally stable time-discretization scheme using a gradient descent implicit in the convex partition and explicit in the concave partition, as given by the form [26], [28], [64]

$$U_{ik}^{n+1} + dt\,\frac{\delta E_{\text{convex}}}{\delta U_{ik}}(U_{ik}^{n+1}) = U_{ik}^n - dt\,\frac{\delta E_{\text{concave}}}{\delta U_{ik}}(U_{ik}^n). \qquad (18)$$

We write this equation in matrix form as

$$\begin{aligned} &\mathbf{U}^{n+1} + dt\left(\epsilon\mathbf{L_s}\mathbf{U}^{n+1} + C\mathbf{U}^{n+1}\right) \\ &\quad = \mathbf{U}^n - dt\left(\frac{1}{2\epsilon}\mathbf{T}^n + \boldsymbol{\mu}(\mathbf{U}^n - \hat{\mathbf{U}}) - C\mathbf{U}^n\right), \end{aligned} \qquad (19)$$

where

$$T_{ik} = \sum_{l=1}^{K} \frac{1}{2} \left(1 - 2\delta_{kl}\right) \left\|\mathbf{u}_i - \boldsymbol{e}_l\right\|_{L_1} \prod_{\substack{m=1 \\ m \neq l}}^{K} \frac{1}{4} \left\|\mathbf{u}_i - \boldsymbol{e}_m\right\|_{L_1}^2, \tag{20}$$

$\boldsymbol{\mu}$ is a diagonal matrix with elements μ_i, and $\hat{\mathbf{U}} = (\hat{\mathbf{u}}_1, \ldots, \hat{\mathbf{u}}_{N_D})^T$.

Solving (19) for $\mathbf{U}^{n+1}$ gives the iteration equation

$$\mathbf{U}^{n+1} = \mathbf{B}^{-1} \left[\left(1 + C\, dt\right) \mathbf{U}^n - \frac{dt}{2\epsilon}\, \mathbf{T}^n - dt\, \boldsymbol{\mu}(\mathbf{U}^n - \hat{\mathbf{U}}) \right] \tag{21}$$

where

$$\mathbf{B} = (1 + C\, dt)\mathbf{I} + \epsilon\, dt\, \mathbf{L_s}. \tag{22}$$

This implicit scheme allows the evolution of $\mathbf{U}$ to be numerically stable regardless of the time step dt, in spite of the numerical "stiffness" of the underlying differential equations which could otherwise force dt to be impractically small.

In general, after the update, the phase field is no longer on the Σ^K simplex. Consequently, we use the procedure in [15] to project back to the simplex.

Computationally, the scheme's numerical efficiency is increased by using a low-dimensional subspace spanned by only a small number of eigenfunctions. Let $\mathbf{X}$ be the matrix of eigenvectors of $\mathbf{L_s}$ and $\boldsymbol{\Lambda}$ be the diagonal matrix of corresponding eigenvalues. We now write $\mathbf{L_s}$ as its eigendecomposition $\mathbf{L_s} = \mathbf{X}\boldsymbol{\Lambda}\mathbf{X}^T$, and set

$$\mathbf{B} = \mathbf{X} \left[(1 + C\, dt)\mathbf{I} + \epsilon\, dt\, \boldsymbol{\Lambda} \right] \mathbf{X}^T, \tag{23}$$

but we approximate $\mathbf{X}$ by a truncated matrix retaining only N_e eigenvectors ($N_e \ll N_D$), to form a matrix of dimension $N_D \times N_e$. The term in brackets is simply a diagonal $N_e \times N_e$ matrix. This allows $\mathbf{B}$ to be calculated rapidly, but more importantly it allows the update step (21) to be decomposed into two significantly faster matrix multiplications (as discussed below), while sacrificing little accuracy in practice.

For initialization, the phase compositions of the fidelity points are set to the vertices of the simplex corresponding to the known labels, while the phase compositions of the rest of the points are set randomly.

The energy minimization proceeds until a steady state condition is reached. The final classes are obtained by assigning class k to node i if $\mathbf{u}_i$ is closest to vertex $\boldsymbol{e}_k$ on the Gibbs simplex. Consequently, the calculation is stopped when

$$\frac{\max_i \left\|{\mathbf{u}_i}^{n+1} - {\mathbf{u}_i}^n\right\|^2}{\max_i \left\|{\mathbf{u}_i}^{n+1}\right\|^2} < \eta, \tag{24}$$

where η represents a given small positive constant.

The algorithm is outlined in Figure 1. While other operator splitting methods have been studied for minimization problems (e.g. [47]), ours has the following advantages: (i) it is direct (i.e. it does not require the solution of further minimization problems), (ii) the resolution can be adjusted by increasing the number of eigenvectors N_e used in the representation of the phase field, and (iii) it has low complexity. To see this final point, observe that each iteration of the multiclass GL algorithm has only $O(N_D K N_e)$ operations for the main loop, since matrix $\mathbf{Z}$ in Figure 1 only has dimensions $N_e \times K$, and then $O(N_D K \log K)$ operations for the projection to the simplex. Usually, $N_e \ll N_D$ and $K \ll N_D$, so the dominant factor is simply the size of the data set N_D. In addition, it is generally the case that the number of iterations required for convergence is moderate (around 50 iterations). Thus, practically speaking, the complexity of the algorithm is linear.

B. Multiclass MBO Reduction

Using the standard Gibbs-simplex Σ^K, the multiclass extension of the algorithm in [49] is straightforward. The notation is the same as in the beginning of the section. While the first step of the algorithm remains the same (except, of course, it is now in matrix form), the second step of the algorithm is modified so that the thresholding is converted to the displacement of the vector field variable towards the closest vertex in the Gibbs simplex. In other words, the row vector ${\mathbf{u}_i}^{n+\frac{1}{2}}$ of step 1 is projected back to the simplex (using the approach outlined in [15] as before) and then a pure phase given by the vertex in the Σ^K simplex closest to ${\mathbf{u}_i}^{n+\frac{1}{2}}$ is assigned to be the new phase composition of node i.

In summary, the new algorithm consists of alternating between the following two steps to obtain approximate solutions $\mathbf{U}^n$ at discrete times:

1) Heat equation with forcing term:

$$\frac{\mathbf{U}^{n+\frac{1}{2}} - \mathbf{U}^n}{dt} = -\mathbf{L_s}\mathbf{U}^n - \boldsymbol{\mu}(\mathbf{U}^n - \hat{\mathbf{U}}). \tag{25}$$

2) Thresholding:

$${\mathbf{u}_i}^{n+1} = \boldsymbol{e}_k, \tag{26}$$

where vertex $\boldsymbol{e}_k$ is the vertex in the simplex closest to $projectToSimplex({\mathbf{u}_i}^{n+\frac{1}{2}})$.

As with the multiclass GL algorithm, when a label is known, it is represented by the corresponding vertex in the Σ^K simplex. The final classification is achieved by assigning node i to class k if if the kth component of $\mathbf{u}_i$ is one. Again, as in the binary case, the diffusion step can be repeated a number of times before thresholding and when that happens, dt is divided by the number of diffusion iterations N_S.

As in the previous section, we use an implicit numerical scheme. For the MBO algorithm, the procedure involves modifying (25) to apply $\mathbf{L_s}$ to $\mathbf{U}^{n+\frac{1}{2}}$ instead of to $\mathbf{U}^n$. This gives the diffusion step

$$\mathbf{U}^{n+\frac{1}{2}} = \mathbf{B}^{-1} \left[\mathbf{U}^n - dt\, \boldsymbol{\mu}(\mathbf{U}^n - \hat{\mathbf{U}}) \right] \tag{27}$$

where

$$\mathbf{B} = \mathbf{I} + dt\, \mathbf{L_s}. \tag{28}$$

As before, we use the eigendecomposition $\mathbf{L_s} = \mathbf{X}\boldsymbol{\Lambda}\mathbf{X}^T$ to write

$$\mathbf{B} = \mathbf{X} \left(\mathbf{I} + dt\, \boldsymbol{\Lambda}\right) \mathbf{X}^T, \tag{29}$$

which we approximate using the first N_e eigenfunctions. The initialization procedure and the stopping criterion are the same as in the previous section.

The multiclass MBO algorithm is summarized in Figure 2. Its complexity is $O(N_D K N_e N_S)$ operations for the main loop, $O(N_D K \log K)$ operations for the projection to the

Fig. 1: Multiclass GL Algorithm

Require: $\epsilon, dt, N_D, N_e, K, \boldsymbol{\mu}, \hat{\mathbf{U}}, \boldsymbol{\Lambda}, \mathbf{X}$
Ensure: out $= \mathbf{U}^{\mathbf{end}}$
  $C \leftarrow \mu + \frac{1}{\epsilon}$
  $\mathbf{Y} \leftarrow [(1 + C\,dt)\mathbf{I} + \epsilon\,dt\boldsymbol{\Lambda}]^{-1}\,\mathbf{X}^T$
  for $i = 1 \rightarrow N_D$ **do**
    $U_{ik}^{\,0} \leftarrow rand((0,1)),\ U_{ik}^{0} \leftarrow projectToSimplex(\mathbf{u}_i{}^0)$. If $\mu_i > 0,\ U_{ik}^{\,0} \leftarrow \hat{U}_{ik}^{\,0}$
  end for
  $n \leftarrow 1$
  while Stop criterion not satisfied **do**
    for $i = 1 \rightarrow N_D$, $k = 1 \rightarrow K$ **do**
      $T_{ik}^{\,n} \leftarrow \sum_{l=1}^{K} \frac{1}{2}\,(1 - 2\delta_{kl})\,\|\mathbf{u}_i{}^n - \boldsymbol{e}_l\|_{L_1} \prod_{m=1, m\neq l}^{K} \frac{1}{4}\,\|\mathbf{u}_i{}^n - \boldsymbol{e}_m\|_{L_1}^2$
    end for
    $\mathbf{Z} \leftarrow \mathbf{Y}\left[(1 + C\,dt)\,\mathbf{U}^n - \frac{dt}{2\epsilon}\,\mathbf{T}^n - dt\,\boldsymbol{\mu}(\mathbf{U}^n - \hat{\mathbf{U}})\right]$
    $\mathbf{U}^{n+1} \leftarrow \mathbf{X}\mathbf{Z}$
    for $i = 1 \rightarrow N_D$ **do**
      $\mathbf{u}_i{}^{n+1} \leftarrow projectToSimplex(\mathbf{u}_i{}^{n+1})$
    end for
    $n \leftarrow n + 1$
  end while

simplex and $O(N_D K)$ operations for thresholding. As in the multiclass GL algorithm, $N_e \ll N_D$ and $K \ll N_D$. Furthermore, N_S needs to be set to three, and due to the thresholding step, we find that extremely few iterations (e.g., 6) are needed to reach steady state. Thus, in practice, the complexity of this algorithm is linear as well, and typical runtimes are very rapid as shown in Table III.

Note that graph analogues of continuum operators, such as gradient and Laplacian, can be constructed using tools of nonlocal discrete calculus. Hence, it is possible to express notions of graph curvature for arbitrary graphs, even with no geometric embedding, but this is not straightforward. For a more detailed discussion about the MBO scheme and motion by mean curvature on graphs, we refer the reader to [59].

IV. Experimental Results

We have tested our algorithms on synthetic data, image labeling, and the MNIST, COIL and WebKB benchmark data sets. In most of these cases, we compute the symmetric normalized graph Laplacian matrix $\mathbf{L}_{\boldsymbol{s}}$, of expression (9), using N-neighborhood graphs: in other words, vertices i and j are connected only if i is among the N nearest neighbors of j or if j is among the N nearest neighbors of i. Otherwise, we set $w(i,j) = 0$. This results in a sparse matrix, making calculations and algorithms more tractable. In addition, for the similarity function we use the local scaling weight function of Zelnik-Manor and Perona [65], defined as

$$w(i,j) = \exp\left(-\frac{d(i,j)^2}{\sqrt{\tau(i)\tau(j)}}\right) \tag{30}$$

where $d(i,j)$ is some distance measure between vertices i and j, such as the L_2 distance, and $\sqrt{\tau(i)} = d(i,k)$ defines a local value for each vertex i, parametrized by M, with k being the index of the Mth closest vertex to i.

With the exception of the image labeling example, all the results and comparisons with other published methods are summarized in Tables I and II. Due to the arbitrary selection of the fidelity points, our reported values correspond to averages obtained over 10 runs with different random selections. The timing results and number of iterations of the two methods are shown in Tables III and IV, respectively. The methods are labeled as "multiclass GL" and "multiclass MBO". These comparisons show that our methods exhibit a performance that is competitive with or better than the current state-of-the-art segmentation algorithms.

Parameters are chosen to produce comparable performance between the methods. For the multiclass GL method, the convexity constant used is: $C = \mu + \frac{1}{\epsilon}$. As described before in expression (17), this is the lower limit that guarantees the convexity and concavity of the terms in the energy partition of the convex splitting strategy employed. For the multiclass MBO method, as discussed in the previous section, the diffusion step can be repeated a number of times before thresholding. In all of our results, we run the diffusion step three times before any thresholding is done ($N_S = 3$).

To compute the eigenvectors and eigenvalues of the symmetric graph Laplacian, we use fast numerical solvers. As we only need to calculate a portion of the eigenvectors to get good results, we compute the eigendecompositions using the Rayleigh-Chebyshev procedure of [1] in all cases except the image labeling example. This numerical solver is especially efficient for producing a few of the smallest eigenvectors of a sparse symmetric matrix. For example, for the MNIST data set of 70,000 images, it was only necessary to calculate 300 eigenvectors, which is less than 0.5% of the data set size. This

Fig. 2: Multiclass MBO Algorithm

Require: $dt, N_D, N_e, N_S, K, \boldsymbol{\mu}, \hat{\mathbf{U}}, \boldsymbol{\Lambda}, \mathbf{X}$
Ensure: out $= \mathbf{U}^{\mathbf{end}}$
  $\mathbf{Y} \leftarrow \left(\mathbf{I} + \frac{dt}{N_S}\boldsymbol{\Lambda}\right)^{-1} \mathbf{X}^T$
  for $i = 1 \rightarrow N_D$ **do**
    $U_{ik}^0 \leftarrow rand((0,1)),\ \mathbf{u}_i{}^0 \leftarrow projectToSimplex(\mathbf{u}_i{}^0)$. If $\mu_i > 0,\ U_{ik}^0 \leftarrow \hat{U}_{ik}^0$
  end for
  $n \leftarrow 1$
  while Stop criterion not satisfied **do**
    for $s = 1 \rightarrow N_S$ **do**
      $\mathbf{Z} \leftarrow \mathbf{Y}\left[\mathbf{U}^n - \frac{dt}{N_S}\boldsymbol{\mu}(\mathbf{U}^n - \hat{\mathbf{U}})\right]$
      $\mathbf{U}^{n+1} \leftarrow \mathbf{XZ}$
    end for
    for $i = 1 \rightarrow N_D$ **do**
      $\mathbf{u}_i{}^{n+1} \leftarrow projectToSimplex(\mathbf{u}_i{}^{n+1})$
      $\mathbf{u}_i{}^{n+1} \leftarrow \boldsymbol{e}_k$, where k is closest simplex vertex to $\mathbf{u}_i{}^{n+1}$
    end for
    $n \leftarrow n + 1$
  end while

is one of the factors that makes our methods very efficient. For the image labeling experiments, we use the Nyström extension method described in [4], [29], [30]. The advantage of the latter method is that it can be efficiently used for very large datasets, because it appoximates the eigenvalues and eigenvectors of a large matrix by calculations done on much smaller matrices formed by randomly chosen parts of the original matrix.

A. Synthetic Data

The synthetic data set we tested our method against is the three moons data set. It is constructed by generating three half circles in $\mathbb{R}^2$. The two half top circles are unit circles with centers at $(0,0)$ and $(3,0)$. The bottom half circle has radius 1.5 and the center at $(1.5, 0.4)$. Five hundred points from each of those three half circles are sampled and embedded in $\mathbb{R}^{100}$ by adding Gaussian noise with standard deviation of 0.14 to each of the 100 components of each embedded point. The dimensionality of the data set, together with the noise, makes segmentation a significant challenge.

The weight matrix of the graph edges was calculated using $N = 10$ nearest neighbors and local scaling based on the 17^{th} closest point ($M = 17$). The fidelity term was constructed by labeling 25 points per class, 75 points in total, corresponding to only 5% of the points in the data set.

The multiclass GL method used the following parameters: 15 eigenvectors, $\epsilon = 1$, $dt = 0.1$, $\mu = 30$, $\eta = 10^{-7}$. The method was able to produce an average of 98.1% of correct classification, with a corresponding computation time of 0.016 s per run on a 2.4 GHz Intel Core i2 Quad without any parallel processing.

Analogously, the multiclass MBO method used the following parameters: 20 eigenvectors, $dt = 0.1$, $\mu = 30$, $\eta = 10^{-7}$. It was able to segment an average of 99.12% of the points correctly over 10 runs with only 3 iterations and about 0.01 s of computation time. One of the results obtained is shown in Figure 3.

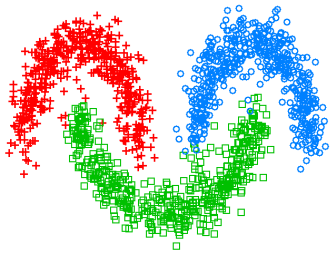


Fig. 3: Segmentation of three moons using multiclass MBO (98.4667% correct).

Table I gives published results from other related methods, for comparison. Note that the results for p-Laplacians [11], Cheeger cuts [57] and binary GL are for the simpler binary problem of two moons (also embedded in $\mathbb{R}^{100}$). While, strictly speaking, these are unsupervised methods, they all incorporate prior knowledge such as a mass balance constraint. We therefore consider them comparable to our SSL approach. The "tree GL" method [31] uses a scalar multiclass GL approach with a tree metric. It can be seen that our methods achieve the highest accuracy on this test problem.

The parameter ϵ determines a scale for the diffuse interface and therefore has consequences in the minimization of the GL energy functional, as discussed in Section II-B. Smaller values of ϵ define a smaller length for the diffuse interface, and at the same time, increasing the relative weight of the potential term with respect to the smoothing term. Therefore, as the parameter ϵ decreases, sharp transitions are generated which in general constitute more accurate classifications. Figure 4

TABLE I: Results for benchmark data sets: Moons, MNIST, COIL and WebKB

Two/Three moons

Method	Accuracy
spectral clustering [31]	80%
p-Laplacian [11]	94%
Cheeger cuts [57]	95.4%
tree GL [31]	97.4%
binary GL [4]	97.7%
multiclass GL	98.1%
multiclass MBO	99.12%

MNIST

Method	Accuracy
p-Laplacian [11]	87.1%
multicut normalized 1-cut [40]	87.64%
linear classifiers [45], [46]	88%
Cheeger cuts [57]	88.2%
boosted stumps [43], [46]	92.3-98.74%
transductive classification [58]	92.6%
tree GL [31]	93.0%
k-nearest neighbors [45], [46]	95.0-97.17%
neural/convolutional nets [17], [45], [46]	95.3-99.65%
nonlinear classifiers [45], [46]	96.4-96.7%
multiclass GL	96.8%
multiclass MBO	96.91%
SVM [21], [45]	98.6-99.32%

COIL

Method	Accuracy
k-nearest neighbors [56]	83.5%
LapRLS [3], [56]	87.8%
sGT [42], [56]	89.9%
SQ-Loss-I [56]	90.9%
MP [56]	91.1%
multiclass GL	91.2%
multiclass MBO	91.46%

WebKB

Method	Accuracy
vector method [12]	64.47%
k-nearest neighbors ($k = 10$) [12]	72.56%
centroid (normalized sum) [12]	82.66%
naive Bayes [12]	83.52%
SVM (linear kernel) [12]	85.82%
multiclass GL	87.2%
multiclass MBO	88.48%

TABLE II: WebKB results with varying fidelity percentage

Method	10%	15%	20%	25%	30%
WebKB results for Multiclass GL (% correct)	81.3%	84.3%	85.8%	86.7%	87.2%
WebKB results for Multiclass MBO (% correct)	83.71%	85.75%	86.81%	87.74%	88.48%

TABLE III: Comparison of timings (in seconds)

Data set	three moons	color images	MNIST	COIL	WebKB
Size	1.5 K	144 K	70 K	1.5 K	4.2 K
Graph Calculation	0.771	0.52	6183.1	0.95	399.35
Eigenvector Calculation	0.331	27.7	1683.5	0.19	64.78
Multiclass GL	0.016	837.1	153.1	0.035	0.49
Multiclass MBO	0.013	40.0	15.4	0.03	0.05

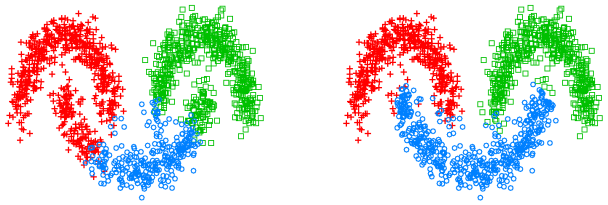
TABLE IV: Comparison of number of iterations

Data set	three moons	color images	MNIST	COIL	WebKB
Multiclass GL	15	770	90	12	20
Multiclass MBO	3	44	7	6	7

compares the performance for two different values of ϵ. Note that the GL results for large ϵ are roughly comparable to those given by a standard spectral clustering approach [31].

B. Co-segmentation

We tested our algorithms on the task of co-segmentation. In this task, two images with a similar topic are used. On one of the images, several regions are labeled. The image labeling task looks for a procedure to transfer the knowledge about regions, specified by the labeled segmentation, onto the unlabeled image. Thus, the limited knowledge about what



(a) $\epsilon = 2.5$ (b) $\epsilon = 1$

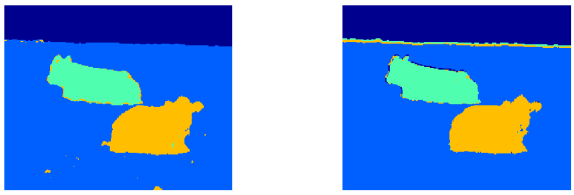
Fig. 4: Three-moons segmentation. Left: $\epsilon = 2.5$ (81.8% correct). Right: $\epsilon = 1$ (97.1 % correct).

defines a region is used to segment similar images without the need for further labelings.

(a) Original Image (b) Labeled Data

Fig. 5: Labeled Color Image

(a) Image to Segment



(b) Multiclass GL (c) Multiclass MBO

Fig. 6: Resulting Color Image Segmentation

On the color image of cows, shown in Figure 5a, some parts of the sky, grass, black cow and red cow have been labeled, as shown in Figure 5b. This is a 319×239 color image. The image to be segmented is a 319×213 color image shown in Figure 6a. The objective is to identify in this second image regions that are similar to the components in the labeled image.

To construct the weight matrix, we use feature vectors defined as the set of intensity values in the neighborhood of a pixel. The neighborhood is a patch of size 5×5. Red, green and blue channels are appended, resulting in a feature vector of dimension 75. A Gaussian similarity graph, as described in equation (7), is constructed with $\sigma = 22$ for both algorithms. Note that for both the labeled and the unlabeled image, nodes that represent similar patches are connected by high-weighted edges, independent of their position within the image. The transfer of information is then enabled through the resulting graph, illustrating the nonlocal characteristics of this unembedded graph-based method.

The eigendecomposition of the Laplacian matrix is approximated using the Nyström method. This involves selecting 250 points randomly to generate a submatrix, whose eigendecomposition is used in combination with matrix completion techniques to generate the approximate eigenvalues for the complete set. Details of the Nyström method are given elsewhere [4], [29], [30]. This approximation drastically reduces the computation time, as seen in Table III.

The multiclass Ginzburg-Landau method used the following parameters: 200 eigenvectors, $\epsilon = 1$, $dt = 0.005$, $\mu = 50$ and $\eta = 10^{-7}$.

The multiclass MBO method used the following parameters: 250 eigenvectors, $dt = 0.005$, $\mu = 300$, $\eta = 10^{-7}$.

One of the results of each of our two methods (using the same fidelity set) is depicted in Figure 6. It can be seen that both methods are able to transfer the identity of all the classes, with slightly better results for mutliclass MBO. Most of the mistakes made correspond to identifying some borders of the red cow as part of the black cow. Multiclass GL also has problems identifying parts of the grass.

C. MNIST Data

The MNIST data set [46] is composed of $70,000$ 28×28 images of handwritten digits 0 through 9. Examples of entries can be found in Figure 7. The task is to classify each of the images into the corresponding digit. The images include digits from 0 to 9; thus, this is a 10 class segmentation problem.

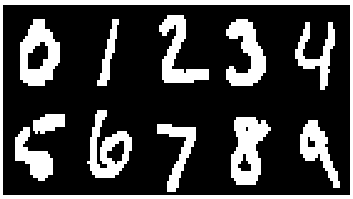


Fig. 7: Examples of digits from the MNIST data base

To construct the weight matrix, we used $N = 8$ nearest neighbors with local scaling based on the 8^{th} closest neighbor ($M = 8$). Note that we perform no preprocessing, i.e. the graph is constructed using the 28×28 images. For the fidelity term, 250 images per class (2500 images corresponding to 3.6% of the data) are chosen randomly.

The multiclass GL method used the following parameters: 300 eigenvectors, $\epsilon = 1$, $dt = 0.15$, $\mu = 50$ and $\eta = 10^{-7}$. The set of 70,000 images was segmented with an average accuracy (over 10 runs) of 96.8% of the digits classified correctly in an average time of 153 s.

The multiclass MBO method used the following parameters: 300 eigenvectors, $dt = 0.15$, $\mu = 50$, $\eta = 10^{-7}$. The algorithm segmented an average of 96.91% of the digits correctly over 10 runs in only 4 iterations and 15.382 s. We display the confusion matrix in Table V. Note that most of the mistakes were in distinguishing digits 4 and 9, and digits 2 and 7.

Table I compares our results with those from other methods in the literature. As with the three moon problem, some of these are based on unsupervised methods but incorporate enough prior information that they can fairly be compared with SSL methods. The methods of linear/nonlinear classifers, k-nearest neighbors, boosted stumps, neural and convolutional nets and SVM are all supervised learning approaches, taking 60,000 of the digits as a training set and 10,000 digits as a testing set [46], in comparison to our SSL approaches where

TABLE V: Confusion Matrix for MNIST Data Segmentation: MBO Scheme

Obtained/True	0	1	2	3	4	5	6	7	8	9
0	6844	20	41	3	3	15	21	1	20	17
1	5	7789	32	8	34	1	14	63	51	14
2	5	22	6731	42	2	4	1	23	19	8
3	0	3	20	6890	1	86	0	1	81	90
4	1	17	6	2	6625	3	7	12	28	67
5	9	0	3	70	0	6077	28	2	109	14
6	31	5	11	3	22	69	6800	0	29	5
7	2	16	117	44	12	9	0	7093	20	101
8	2	2	21	46	4	17	5	2	6398	22
9	4	3	8	33	121	32	0	96	70	6620

we take only 3.6% of the points for the fidelity term. Our algorithms are nevertheless competitive with, and in most cases outperform, these supervised methods. Moreover, we perform no preprocessing or initial feature extraction on the image data, unlike most of the other methods we compare with (we exclude from the comparison the methods that deskewed the image). While there is a computational price to be paid in forming the graph when data points use all 784 pixels as features (see graph calculation time in Table III), this is a one-time operation that conceptually simplifies our approach.

D. COIL dataset

We evaluated our performance on the benchmark COIL data set [14], [52]. This is a set of color 128×128 images of 100 objects, taken at different angles. The red channel of each image was then downsampled to 16×16 pixels by averaging over blocks of 8×8 pixels. Then 24 of the objects were randomly selected and then partitioned into six classes. Discarding 38 images from each class leaves 250 per class, giving a data set of 1500 data points.

To construct the weight matrix, we used $N = 4$ nearest neighbors with local scaling based on the 4^{th} closest neighbor ($M = 4$). The fidelity term was constructed by labeling 10% of the points, selected at random.

For multiclass GL, the parameters were: 35 eigenvectors, $\epsilon = 1$, $dt = 0.05$, $\mu = 50$ and $\eta = 10^{-7}$. This resulted in 91.2% of the points classified correctly (average) in 0.035 s.

For multiclass MBO, the parameters were: 50 eigenvectors, $dt = 0.2$, $\mu = 100$, $\eta = 10^{-7}$. We obtained an accuracy of 91.46%, averaged over 10 runs. The procedure took 6 iterations and 0.03 s.

Comparative results reported in [56] are shown in Table I. These are all SSL methods (with the exception of k-nearest neighbors which is supervised), using 10% fidelity just as we do. Our results are of comparable or greater accuracy.

E. WebKB dataset

Finally, we tested our methods on the task of text classification on the WebKB data set [20]. This is a collection of webpages from Cornell, Texas, Washington and Wisconsin universities, as well as other miscellaneous pages from other universities. The webpages are to be divided into four classes: project, course, faculty and student. The data set is preprocessed as described in [12].

To construct the weight matrix, we used 575 nearest neighbors. Tfidf term weighting [12] is used to represent the website feature vectors. They were then normalized to unitary length. The weight matrix points are calculated using cosine similarity.

For the multiclass GL, the parameters were: 250 eigenvectors, $\epsilon = 1$, $dt = 1$, $\mu = 50$ and $\eta = 10^{-7}$. The average accuracies obtained for fidelity sets of different sizes are given in Table II. The average computation time was 0.49 s.

For the multiclass MBO, the parameters were: 250 eigenvectors, $dt = 1$, $\mu = 4$, $\eta = 10^{-7}$. The average accuracies obtained for fidelity sets of different sizes are given in Table II. The procedure took an average of 0.05 s and 7 iterations.

We compare our results with those of several supervised learning methods reported in [12], shown in Table I. For these methods, two-thirds of the data were used for training, and one third for testing. Our SSL methods obtain higher accuracy, using only 20% fidelity (for multiclass MBO). Note that a larger sample of points for the fidelity term reduces the error in the results, as shown in Table II. Nevertheless, the accuracy is high even for the smallest fidelity sets. Therefore, the methods appear quite adequate for the SSL setting where only a few labeled data points are known beforehand.

Multiclass GL and MBO: All the results reported point out that both multiclass GL and multiclass MBO perform well in terms of data segmentation accuracy. While the ability to tune multiclass GL can be an advantage, multiclass MBO is simpler and, in our examples, displays even better performance in terms of its greater accuracy and the fewer number of iterations required. Note that even though multiclass GL leads to the minimization of a non-convex function, in practice the results are comparable with other convex TV-based graph methods such as [9]. Exploring the underlying connections of the energy evolution of these methods and the energy landscape for the relaxed Cheeger cut minimization recently established in [8] are to be explored in future work.

V. Conclusions

We have presented two graph-based algorithms for multiclass classification of high-dimensional data. The two algorithms are based on the diffuse interface model using the Ginzburg-Landau functional, and the multiclass extension is

obtained using the Gibbs simplex. The first algorithm minimizes the functional using gradient descent and a convex-splitting scheme. The second algorithm executes a simple scheme based on an adaptation of the classical numerical MBO method. It uses fewer parameters than the first algorithm, and while this may in some cases make it more restrictive, in our experiments it was highly accurate and efficient.

Testing the algorithms on synthetic data, image labeling and benchmark data sets shows that the results are competitive with or better than some of the most recent and best published algorithms in the literature. In addition, our methods have several advantages. First, they are simple and efficient, avoiding the need for intricate function minimizations or heavy preprocessing of data. Second, a relatively small proportion of fidelity points is needed for producing an accurate result. For most of our data sets, we used at most 10% of the data points for the fidelity term; for synthetic data and the two images, we used no more than 5%. Furthermore, as long as the fidelity set contains samples of all classes in the problem, a random initialization is enough to produce good multiclass segmentation results. Finally, our methods do not use one-vs-all or sequences of binary segmentations that are needed for some other multiclass methods. We therefore avoid the bias and extra processing that is often inherent in those methods.

Our algorithms can take advantage of the sparsity of the neighborhood graphs generated by the local scaling procedure of Zelnik-Manor and Perona [65]. A further reason for the strong practical performance of our methods is that the minimization equations use only the graph Laplacian, and do not contain divergences or any other first-order derivative terms. This allows us to use rapid numerical methods. The Laplacian can easily be inverted by projecting onto its eigenfunctions, and in practice, we only need to keep a small number of these. Techniques such as the fast numerical Rayleigh-Chebyshev method of Anderson [1] are very efficient for finding the small subset of eigenvalues and eigenvectors needed. In certain cases, we obtain additional savings in processing times by approximating the eigendecomposition of the Laplacian matrix through the Nyström method [4], [29], [30], which is effective even for very large matrices: we need only compute a small fraction of the weights in the graph, enabling the approximation of the eigendecomposition of a fully connected weight matrix using computations on much smaller matrices.

Thus, there is a significant computational benefit in not having to calculate any first-order differential operators. In view of this, we have found that for general graph problems, even though GL requires minimizing a non-convex functional, the results are comparable in accuracy to convex TV-based graph methods such as [9]. For MBO, the results are similarly accurate, with the further advantage that the algorithm is very rapid. We note that for other problems such as in image processing that are suited to a continuum treatment, convex methods and maxflow-type algorithms are in many cases the best approach [13], [62]. It would be very interesting to try to extend our gradient-free numerical approach to graph-based methods that directly use convex minimization, such as the method described in [63].

Finally, comparatively speaking, multiclass MBO performed better than multiclass GL in terms of accuracy and convergence time for all of the data sets we have studied. Nevertheless, we anticipate that more intricate geometries could impair its effectiveness. In those cases, multiclass GL might still perform well, due to the additional control provided by tuning ϵ to increase the thickness of the interfaces, producing smoother decision functions.

ACKNOWLEDGMENTS

The authors are grateful to the reviewers for their comments and suggestions, which helped to improve the quality and readability of the manuscript. In addition, the authors would like to thank Chris Anderson for providing the code for the Rayleigh-Chebyshev procedure of [1]. This work was supported by ONR grants N000141210838, N000141210040, N0001413WX20136, AFOSR MURI grant FA9550-10-1-0569, NSF grants DMS-1118971 and DMS-0914856, the DOE Office of Science's ASCR program in Applied Mathematics, and the W. M. Keck Foundation. Ekaterina Merkurjev is also supported by an NSF graduate fellowship.

REFERENCES

[1] C. Anderson, "A Rayleigh-Chebyshev procedure for finding the smallest eigenvalues and associated eigenvectors of large sparse Hermitian matrices," *J. Comput. Phys.*, vol. 229, pp. 7477–7487, 2010.

[2] G. Barles and C. Georgelin, "A simple proof of convergence for an approximation scheme for computing motions by mean curvature," *SIAM Journal on Numerical Analysis*, vol. 32, no. 2, pp. 484–500, 1995.

[3] M. Belkin, P. Niyogi, and V. Sindhwani, "Manifold regularization: A geometric framework for learning from labeled and unlabeled examples," *The Journal of Machine Learning Research*, vol. 7, pp. 2399–2434, 2006.

[4] A. Bertozzi and A. Flenner, "Diffuse interface models on graphs for classification of high dimensional data," *Multiscale Modeling & Simulation*, vol. 10, no. 3, pp. 1090–1118, 2012.

[5] A. Bertozzi and Y. van Gennip, "Γ-convergence of graph Ginzburg-Landau functionals," *Advances in Differential Equations*, vol. 17, no. 11–12, pp. 1115–1180, 2012.

[6] A. Bertozzi, S. Esedoğlu, and A. Gillette, "Inpainting of binary images using the Cahn-Hilliard equation," *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 285–291, 2007.

[7] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222 –1239, 2001.

[8] X. Bresson, T. Laurent, D. Uminsky, and J. H. von Brecht, "Convergence and energy landscape for Cheeger cut clustering," *Advances in Neural Information Processing Systems*, 2012.

[9] ——, "Multiclass total variation clustering," 2013. [Online]. Available: http://arxiv.org/abs/1306.1185

[10] X. Bresson, X.-C. Tai, T. F. Chan, and A. Szlam, "Multi-class transductive learning based on ℓ^1 relaxations of Cheeger cut and Mumford-Shah-Potts model," *UCLA CAM Report 12-03*, 2012.

[11] T. Bühler and M. Hein, "Spectral clustering based on the graph p-Laplacian," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 81–88.

[12] A. Cardoso, "Datasets for single-label text categorization." [Online]. Available: http://www.ist.utl.pt/~acardoso/datasets/

[13] A. Chambolle, D. Cremers, and T. Pock, "A convex approach to minimal partitions," *SIAM J. Imaging Sciences*, vol. 5, no. 4, pp. 1113–1158, 2012.

[14] O. Chapelle, B. Schölkopf, and A. Zien, Eds., *Semi-Supervised Learning*. Cambridge, MA: MIT Press, 2006. [Online]. Available: http://www.kyb.tuebingen.mpg.de/ssl-book

[15] Y. Chen and X. Ye, "Projection onto a simplex," *arXiv preprint arXiv:1101.6081*, 2011.

[16] F. Chung, *Spectral Graph Theory*. American Mathematical Society, 1997, vol. 92.

[17] D. Cireşan, U. Meier, J. Masci, L. Gambardella, and J. Schmidhuber, "Flexible, high performance convolutional neural networks for image classification," in *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two*. AAAI Press, 2011, pp. 1237–1242.

[18] C. Couprie, L. Grady, L. Najman, and H. Talbot, "Power watershed: A unifying graph-based optimization framework," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 7, pp. 1384–1399, 2011.

[19] C. Couprie, L. Grady, H. Talbot, and L. Najman, "Combinatorial continuous maximum flow," *SIAM Journal on Imaging Sciences*, vol. 4, no. 3, pp. 905–930, 2011.

[20] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, "Learning to extract symbolic knowledge from the world wide web," in *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*. AAAI Press, 1998, pp. 509–516. [Online]. Available: http://www.cs.cmu.edu/˜webkb

[21] D. Decoste and B. Schölkopf, "Training invariant support vector machines," *Machine Learning*, vol. 46, no. 1, pp. 161–190, 2002.

[22] T. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *arXiv preprint cs/9501101*, 1995.

[23] J. A. Dobrosotskaya and A. L. Bertozzi, "A wavelet-Laplace variational technique for image deconvolution and inpainting," *IEEE Transactions on Image Processing*, vol. 17, no. 5, pp. 657–663, 2008.

[24] ——, "Wavelet analogue of the Ginzburg-Landau energy and its Γ-convergence," *Interfaces and Free Boundaries*, vol. 12, no. 2, pp. 497–525, 2010.

[25] A. Elmoataz, O. Lezoray, and S. Bougleux, "Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing," *IEEE Transactions on Image Processing*, vol. 17, no. 7, pp. 1047–1060, 2008.

[26] S. Esedoḡlu and Y. Tsai, "Threshold dynamics for the piecewise constant Mumford–Shah functional," *Journal of Computational Physics*, vol. 211, no. 1, pp. 367–384, 2006.

[27] L. C. Evans, "Convergence of an algorithm for mean curvature motion," *Indiana University Mathematics Journal*, vol. 42, no. 2, pp. 533–557, 1993.

[28] D. J. Eyre, "An unconditionally stable one-step scheme for gradient systems," http://www.math.utah.edu/˜eyre/research/methods/papers.html, 1998.

[29] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nyström method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 26, no. 2, pp. 214–225, 2004.

[30] C. Fowlkes, S. Belongie, and J. Malik, "Efficient spatiotemporal grouping using the Nyström method," in *In Proc. IEEE Conf. Comput. Vision and Pattern Recognition*, 2001, pp. 231–238.

[31] C. Garcia-Cardona, A. Flenner, and A. G. Percus, "Multiclass diffuse interface models for semi-supervised learning on graphs," in *Proceedings of the 2th International Conference on Pattern Recognition Applications and Methods*. SciTePress, 2013.

[32] H. Garcke, B. Nestler, B. Stinner, and F. Wendler, "Allen-Cahn systems with volume constraints," *Mathematical Models and Methods in Applied Sciences*, vol. 18, no. 8, 2008.

[33] G. Gilboa and S. Osher, "Nonlocal operators with applications to image processing," *Multiscale Modeling & Simulation*, vol. 7, no. 3, pp. 1005–1028, 2008.

[34] T. Goldstein and S. Osher, "The split Bregman method for L_1-regularized problems," *SIAM Journal on Imaging Sciences*, vol. 2, no. 2, pp. 323–343, 2009.

[35] L. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. Springer, 2010.

[36] L. Grady, "Multilabel random walker image segmentation using prior models," in *Proceedings of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Vol. 1*, 2005, pp. 763–770.

[37] ——, "Random walks for image segmentation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 28, no. 11, pp. 1768–1783, 2006.

[38] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann, "Random walks for interactive alpha-matting," in *VIIP*, 2005.

[39] T. Hastie and R. Tibshirani, "Classification by pairwise coupling," *The Annals of Statistics*, vol. 26, no. 2, pp. 451–471, 1998.

[40] M. Hein and S. Setzer, "Beyond spectral clustering - tight relaxations of balanced graph cuts," in *Advances in Neural Information Processing Systems 24*, J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 2366–2374.

[41] H. Hu, T. Laurent, M. A. Porter, and A. L. Bertozzi, "A method based on total variation for network modularity optimization using the MBO scheme," *SIAM J. Appl. Math.*, 2013. [Online]. Available: http://arxiv.org/abs/1304.4679

[42] T. Joachims *et al.*, "Transductive learning via spectral graph partitioning," in *International Conference on Machine Learning*, vol. 20, no. 1, 2003, p. 290.

[43] B. Kégl and R. Busa-Fekete, "Boosting products of base classifiers," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 497–504.

[44] R. Kohn and P. Sternberg, "Local minimizers and singular perturbations," *Proc. Roy. Soc. Edinburgh Sect. A*, vol. 111, no. 1-2, pp. 69–84, 1989.

[45] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[46] Y. LeCun and C. Cortes, "The MNIST database of handwritten digits." [Online]. Available: http://yann.lecun.com/exdb/mnist/

[47] J. Lellmann, J. H. Kappes, J. Yuan, F. Becker, and C. Schnörr, "Convex multi-class image labeling by simplex-constrained total variation," IWR, University of Heidelberg, Technical Report, October 2008. [Online]. Available: http://www.ub.uni-heidelberg.de/archiv/8759/

[48] A. Levin, A. Rav-Acha, and D. Lischinski, "Spectral matting," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1699 –1712, 2008.

[49] E. Merkurjev, T. Kostic, and A. L. Bertozzi, "An mbo scheme on graphs for classification and image processing," *SIAM Journal on Imaging Sciences*, vol. 6, no. 4, pp. 1903–1930, 2013.

[50] B. Merriman, J. K. Bence, and S. J. Osher, "Motion of multiple junctions: a level set approach," *J. Comput. Phys.*, vol. 112, no. 2, pp. 334–363, 1994. [Online]. Available: http://dx.doi.org/10.1006/jcph.1994.1105

[51] B. Mohar, "The Laplacian spectrum of graphs," *Graph Theory, Combinatorics, and Applications*, vol. 2, pp. 871–898, 1991.

[52] S. Nene, S. Nayar, and H. Murase, "Columbia Object Image Library (COIL-100)," *Technical Report CUCS-006-96*, 1996. [Online]. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil-100.php

[53] J. Rubinstein, P. Sternberg, and J. Keller, "A simple proof of convergence for an approximation scheme for computing motions by mean curvature," *SIAM Journal of Applied Mathematics*, vol. 49, no. 1, pp. 116–133, 1989.

[54] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000.

[55] B. Simons, *Phase Transitions and Collective Phenomena*, http://www.tcm.phy.cam.ac.uk/ bds10/phase.html, University of Cambridge, 1997.

[56] A. Subramanya and J. Bilmes, "Semi-supervised learning with measure propagation," *Journal of Machine Learning Research*, vol. 12, pp. 3311–3370, 2011.

[57] A. Szlam and X. Bresson, "Total variation and Cheeger cuts," in *Proceedings of the 27th International Conference on Machine Learning*, J. Fürnkranz and T. Joachims, Eds. Haifa, Israel: Omnipress, 2010, pp. 1039–1046.

[58] A. D. Szlam, M. Maggioni, and R. R. Coifman, "Regularization on graphs with function-adapted diffusion processes," *Journal of Machine Learning Research*, vol. 9, pp. 1711–1739, 2008.

[59] Y. van Gennip, N. Guillen, B. Osting, and A. L. Bertozzi, "Mean curvature, threshold dynamics, and phase field theory on finite graphs," 2013. [Online]. Available: http://www.math.ucla.edu/˜bertozzi/papers/graph_curve.pdf

[60] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[61] J. Wang, T. Jebara, and S. Chang, "Graph transduction via alternating minimization," in *Proceedings of the 25th international conference on Machine learning*. Citeseer, 2008, pp. 1144–1151.

[62] J. Yuan, E. Bae, and X.-C. Tai, "A study on continuous max-flow and min-cut approaches," in *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010, pp. 2217–2224.

[63] J. Yuan, E. Bae, X.-C. Tai, and Y. Boykov, "A fast continuous max-flow approach to potts model," in *Computer Vision–ECCV 2010*. Springer, 2010, pp. 379–392.

[64] A. L. Yuille and A. Rangarajan, "The concave-convex procedure (CCCP)," *Neural Computation*, vol. 15, no. 4, pp. 915–936, 2003.

[65] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," *Advances in Neural Information Processing Systems*, vol. 17, pp. 1601–1608, 2004.

[66] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in Neural Information*

Processing Systems 16, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. Cambridge, MA: MIT Press, 2004, pp. 321–328.

[67] D. Zhou and B. Schölkopf, "A regularization framework for learning from graph data," in *Workshop on Statistical Relational Learning*. Banff, Canada: International Conference on Machine Learning, 2004.

[68] X. Zhu, "Semi-supervised learning literature survey," University of Wisconsin-Madison, Technical Report 1530, Computer Sciences, 2005.

Cristina Garcia-Cardona is a Postdoctoral Fellow at Claremont Graduate University. She obtained her Bachelor's degree in Electrical Engineering from Universidad de Los Andes in Colombia and her Master's degree in Emergent Computer Sciences from Universidad Central de Venezuela. Recently, she received her PhD in Computational Science from the Claremont Graduate University and San Diego State University joint program, working under the supervision of Prof. Allon Percus and Dr. Arjuna Flenner. Her research interests include energy minimization and graph algorithms.

Ekaterina Merkurjev is a fourth year graduate student at the UCLA Department of Mathematics. She obtained her Bachelors and Masters degrees in Applied Mathematics from UCLA in 2010. She is currently working on a PhD under the supervision of Prof. Andrea Bertozzi. Her research interests include image processing and segmentation.

Andrea L. Bertozzi received the BA, MA, and PhD degrees in mathematics from Princeton University, Princeton, NJ, in 1987, 1988, and 1991 respectively. She was on the faculty of the University of Chicago, Chicago, IL, from 1991-1995 and Duke University, Durham, NC, from 1995-2004. During 1995-1996, she was the Maria Goeppert-Mayer Distinguished Scholar at Argonne National Laboratory. Since 2003, she has been with the University of California, Los Angeles, as a Professor of Mathematics and currently serves as the Director of Applied Mathematics. In 2012 she was appointed the Betsy Wood Knapp Chair for Innovation and Creativity. Her research interests include image inpainting, image segmentation, cooperative control of robotic vehicles, swarming, and fluid interfaces, and crime modeling. Prof. Bertozzi is a Fellow of both the Society for Industrial and Applied Mathematics and the American Mathematical Society; she is a member of the American Physical Society. She has served as a Plenary/Distinguished Lecturer for both SIAM and AMS and is an Associate Editor for the SIAM journals Multiscale Modelling and Simulation, Mathematical Analysis. She also serves on the editorial board of Interfaces and Free Boundaries, Applied Mathematics Research Express, Nonlinearity, Appl. Math. Lett., Math. Mod. Meth. Appl. Sci. (M3AS), J. Nonlinear Sci, J. Stat. Phys., Comm. Math. Sci., Nonlinear Anal. Real World Appl., and Adv. Diff. Eq. Her past honors include a Sloan Foundation Research Fellowship, the Presidential Career Award for Scientists and Engineers, and the SIAM Kovalevsky Prize in 2009.

Arjuna Flenner received his Ph.D. in Physics at the University of Missouri-Columbia in 2004. His major emphasis was mathematical physics. Arjuna Flenner's research interests at the Naval Air Weapons Centre at China Lake include image processing, machine learning, statistical pattern recognition, and computer vision. In particular, he has investigated automated image understanding algorithms for advanced naval capabilities. His main research areas are nonlocal operators, geometric diffusion, graph theory, non-parametric Bayesian analysis, and a-contrario hypothesis testing methods. Arjuna Flenner was a US Department of Energy GAANN Fellowship in 1997-2001, and currently is also a visiting research professor at Claremont Graduate University.

Allon G. Percus received his BA in physics from Harvard in 1992 and his PhD from the Université Paris-Sud, Orsay in 1997. He was a member of the scientific staff at Los Alamos National Laboratory in the Division of Computer and Computational Sciences, and from 2003 to 2006 he served as Associate Director of the Institute for Pure and Applied Mathematics at UCLA. Since 2009, he has been Associate Professor of Mathematics at Claremont Graduate University. His research interests combine discrete optimization, combinatorics and statistical physics, exploiting physical models and techniques to study the performance of algorithms on NP-hard problems.